Anonymous Authentication for Smartcards

Jan HAJNÝ

Dept. of Telecommunications, Brno University of Technology, Purkynova 118, 612 00 Brno, Czech Republic

hajny@feec.vutbr.cz

Abstract. *The paper presents an innovative solution in the field of RFID (Radio-Frequency IDentification) smartcard authentication. Currently the smartcards are used for many purposes - e.g. employee identification, library cards, student cards or even identity credentials. Personal identity is revealed to untrustworthy entities every time we use these cards. Such information could later be used without our knowledge and for harmful reasons like shopping pattern scanning or even movement tracking. We present a communication scheme for keeping one's identity private in this paper. Although our system provides anonymity, it does not allow users to abuse this feature. The system is based on strong cryptographic primitives that provide features never available before. Besides theoretical design of the anonymous authentication scheme and its analysis we also provide implementation results.*

Keywords

RFID, privacy, anonymity, cryptography, authentication, smartcards.

1. Current State Analysis

The authentication scheme presented in this paper is designed to solve the problem of private data leak during smartcard use. Currently we experience a heavy expansion of varied smartcards and their use in an everyday life. We use them automatically when we arrive to or leave work, buy meals in canteens, shop in supermarkets or rent videos. It is obvious that data about smartcard use are stored somewhere. It would not be too difficult to find out our eating patterns, shopping patterns or information about our movement if we have a proper access to those data. What is more important is that they could be used by companies to create even more focused advertisement based on information about us. By gathering of all our data we lose our privacy and our identity could be misused in many ways. We can take many examples from [20], [21] to illustrate such a loss of privacy.

If we analyze the above mentioned services we may realize that the identity revelation is not necessary in most cases. The service provider needs to know that a client is an allowed user (e.g. who paid some fee) but he does not have to know the concrete identity until the service is abused. That is a feature we provide in our scheme - the user will stay anonymous during the service use as long as he adheres to the rules.

For anonymous authentication we must be able to provide a protocol which will verify user's eligibility to use the service without his identity revelation. This could be done by the verification of a group membership while the group is the sum of all users allowed to use a service. The concept of group signatures introduced in [7] can be used for this task. If we are successful in creation of such a protocol the service provider will be able to decide whether a client is a member of the group of valid users but no more information will be released about the user. But such a protocol would be useless in practice because there would be no responsibility for one's behavior as all users would be totally anonymous. There must be a revelation mechanism to provide a tool for the identity revelation in the case of service abuse. A similar mechanism is used for the detection of double spenders in e-cash systems like [1]. We will adapt this feature to our scheme to provide user responsibility.

For the smartcard environment the communication must be very efficient to be usable in practice. Our scheme works with a communication-computation tradeoff as most of resource demanding calculations are run on computers only. For the authentication phase itself we use primitive calculations done by a .NET programmable smartcard. Such a tradeoff was possible thank to the use of very efficient yet cryptographically secure Σ-protocols.

1.1 Related Work

Our goal is to provide a solution to the problem of user privacy because we are convinced that there are no practical smartcard anonymous authentication systems available to this date. Nevertheless there are some concepts used in computer networks which can be taken as a starting point. The most promising ones are group signatures, e-cash systems and credential systems.

Group signature schemes were introduced in [7] to preserve signer's anonymity in a similar way as our anonymous authentication scheme does. In practice there is a verifiable group signature made by a user belonging to an established group but the signer's identity remains hidden until the revocation is needed. In that case a trusted revocation manager

is called to "open" the signature and reveal the signer. There has been an intensive research in the area of group signatures. The first schemes were practically inefficient as signatures were too long and dependent on a group size. The more advanced ones like [6] improved the efficiency but still remained off the practical use. In a recent time very efficient schemes were published [13], [10]. Some recent group signatures [12], [15], [23] also address the problem of spread revocation. Those split the revocation manager into more entities. Such a feature was also put to our solution. Nevertheless only a minority of the above mentioned schemes is usable in the smartcard environment which is very sensitive to the right communication-computation tradeoff. That is the reason why we designed our scheme with very limited computations on a smartcard itself.

The offline e-cash systems [4], [5], [18] were analyzed during the work on the authentication system published in this article. The attractive property of a double spender detection was adapted to revoke unwanted users in our scheme. The core of an authentication protocol has been adapted from Brand's e-cash scheme [1] which is very efficient.

The last group of the related systems is represented by credential systems. The most advanced scheme to our knowledge [3] was examined to find similar properties. The purpose of our scheme is different (we need only authentication without the work with credentials) and the scheme is more robust and complex as well as lacking some important features. For our purpose the identity spread revelation (and revocation) is necessary for the service to be practically usable. We find such a feature almost missing in credential systems.

We also analyzed already published anonymous authentication systems with functionality similar to our scheme. Schemes [19], [16] are not usable for our purpose because of the need for trusted HW, scheme [2] works with tags only. Other schemes like [14] need a trusted third party to be able to reveal user's identity.

1.2 Our Contribution

The anonymous authentication scheme presented in this paper was specifically designed to be used in a smartcard environment. That is the reason why all resource demanding computations are done on the computer side preserving smartcard resources. We used a tradeoff to pre-compute all necessary values to make the authentication phase fast and efficient. The scheme still fulfills all security requirements as it provides user anonymity, exculpability, no-framing, sound and complete authentication, user responsibility and revocation with tracing if needed.

In contrast to many schemes, our system works without any trusted entity/HW and is completely independent on the user group size. The only parameter determining the amount of communication data is the security parameter.

2. Requirements

We already identified some basic scheme requirements in the previous chapter. We make these demands more formal in this section. We can adapt requirements for group signatures stated in [17] as they serve a very similar purpose on the computer network platform:

1. Soundness and Completeness: valid users are always accepted to use a service while invalid ones are always rejected.

2. Anonymity: no one can learn the identity of a user based on the authentication information released by the protocol (until system rules are broken).

3. Spread Traceability: there must be a way to reveal user identity based on authentication information released by the protocol but only if rules are broken by that user. Such an identity revelation cannot be done by a single entity but only by joint cooperation of exactly specified entities.

4. No Framing: no-one (not even all other valid users) can forge false authentication information which would frame a user for a non-existing service use.

By the implementation of these requirements we get a scheme which allows only valid users to use the service and denies the rest (Requirement 1). Even if all communication data leak to the advertiser the user will stay private and anonymous. No more information other than eligibility to use the service will be leaked. Even the service provider is unable to identify the user (Requirement 2). We prevent users to abuse their anonymity by the Requirement 3 because any user who broke the system rules would be uniquely determined. We also prevent users to falsely frame other valid users by the Requirement 4.

3. Scheme Design

The communication pattern is given by four entities involved in the authentication process. We expect a client who wants to use a service, an Authentication Server (AS) which manages valid users, an access device which accepts/denies users to use the service and finally a Public Authority (PA) which is used for disputes. The principle is based on tokens which are used as tickets for services. The user who has a valid token and knows its construction is allowed to use the service. The token is created by the user based on information given by both AS and PA. The communication pattern is depicted in Fig. 1.

Before we present the anonymous authentication scheme in detail we must introduce some cryptographic primitives. All of them are the outcomes of modern cryptography and altogether they create a unique core which allows us to fulfill all requirements stated in Section 2.

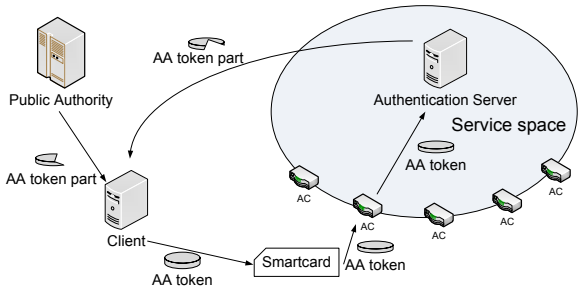


Fig. 1. The communication pattern of the scheme.

3.1 Used Cryptographic Primitives

Σ-protocols

The Σ-protocols (or Sigma-protocols) [8] are products of modern cryptography which can be used as building blocks for many purposes. Nevertheless they are used as proofs between a Prover (P) and a Verifier (V) in most common scenarios. We have chosen these protocols because of their effectiveness and security. For the Σ-protocol to be formerly correct these requirements must be fulfilled:

- 3-way pattern: there are 3 messages sent. The first one goes from the Prover to the Verifier , the second vice versa and the last one again from the Prover to the Verifier. Besides these messages some environment must be set (generators g_1, g_2 and a cyclic multiplicative group in our case).
- Completeness: an honest Prover must be always successful (i.e. accepted by V).
- Special Soundness: a cheating Prover must be able to answer at most one challenge in the protocol.
- HVSZK (Honest Verifier Special Zero-Knowledge): the protocol releases no extra (unintended) information from its run with an honest Verifier.

We stated these requirements in a very informal way which is sufficient for our purpose but a detailed specification could be found e.g. here [9]. The Schnorr protocol [22] is q practical example of the Σ-protocol. We have chosen this protocol because of its effectiveness (it could be used for a computation-communication tradeoff) and flexibility.

Proof of Knowledge of Discrete Logarithm

Schnorr's protocol can be used as the Proof of Knowledge of Discrete Logarithm (PKDL) where the Prover proves to the Verifier that he knows a certain discrete logarithm without its revelation. The Verifier learns only whether the Prover knows the discrete log or not; nothing else from this protocol.

The protocol works with the subgroup $\mathbb{Z}_p$ of the cyclic multiplicative group $\mathbb{Z}_n$ where n is a secure prime modulus, p a high prime divisor of $n-1$ and g an element of order p in $\mathbb{Z}_n$. These variables are pre-shared to entities before an actual use. The Prover knows a secret exponent w (which works as a private identity key because it is not released by the protocol to anyone, V included). P chooses $r \in_R \mathbb{Z}_p$ and sends $c = g^w \bmod n, a = g^r \bmod n$ in the first move. V replies with a random challenge $e \in_R \mathbb{Z}_p$. The Verifier accepts P if he is able to send the answer of the form $z = ew + r \bmod p$ as the last move. This is checked by V by the evaluation of the equation $g^z \equiv ac^e \pmod n$. It can be formally shown that this protocol fulfills all requirements for a Σ-protocol. We can use it to prove the knowledge of a secret key without exposing it to anyone. Even more, we can use it to convince someone that we made computations correctly and that we used the key inside further constructions. The communication is illustrated by Fig. 2.

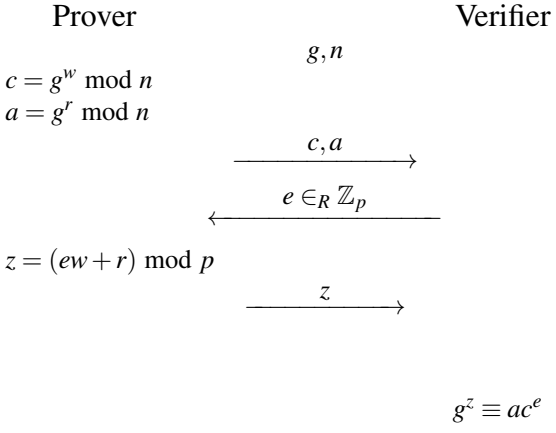


Fig. 2. Proof of Knowledge of Discrete Logarithm.

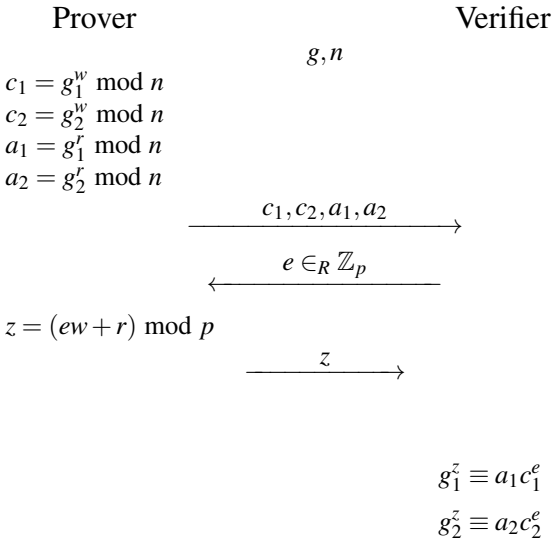


Fig. 3. Proof of Discrete Logarithm Equivalence.

Proof of Discrete Logarithm Equivalence

The second use of the protocol is the Proof of Discrete Logarithm Equivalence (PDLE). We can prove the construction of two numbers c_1, c_2 which were created by the exponentiation of different roots by the same exponent in $\bmod n$. This exponent is not revealed during the run of the protocol. The protocol works in a very similar fashion as the PKDL - see Fig. 3.

We can run this protocol also non-interactively (NIPDLE) by choosing the challenge e as a hash of values c and a. With these primitives we can build the scheme now.

3.2 Anonymous Authentication Scheme

Our authentication scheme is divided into two phases - registration and authentication. The user pays for the service and registers at AS during the registration phase. AS then provides the user with partial information needed to create an authentication token. With this information the user must contact PA which provides the rest needed for a valid token creation. The registration phase is run only through computer networks and could be run as many times as necessary (to get as many tokens as necessary). As soon as the user has the desired number of authentication tokens he can transfer them from the computer to his smartcard. The authentication phase follows. The whole authentication phase runs between the smartcard and an access device, no communication with client computer is necessary. The scheme is illustrated in Fig. 4.

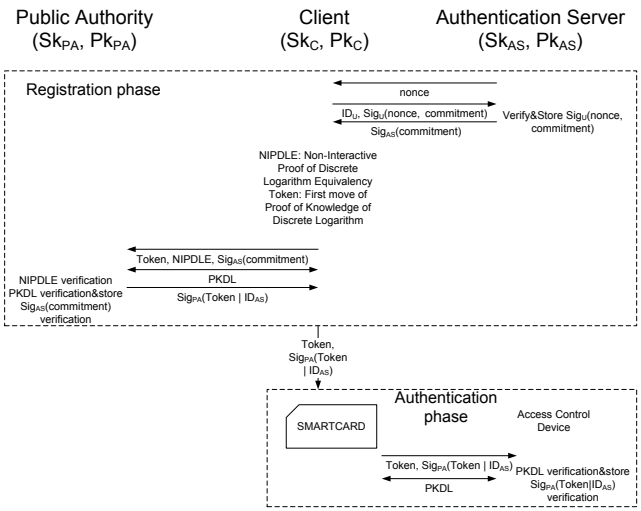


Fig. 4. Authentication scheme.

Registration Phase

The purpose of this phase is to create a token usable for authentication. The token is a construction which must include user identity but only in a hidden form inaccessible by anyone but the user. The construction must release the identity in the case of rule breaking. We use the first step (values c and a) of the above mentioned PKDL protocol as the token because the protocol can work for identity hiding (identity is used as a secret exponent) as well as for authentication (only valid users know the exponent). We will also use the feature of Σ-protocols - they release the secret exponent in the case of multiple runs. The registration phase works as depicted in Fig. 4.

All entities have public cryptography keypairs (Pk, Sk). The environment (safe prime modulus n for the group, p for the subgroup and g_1, g_2 for its generators) is preshared. The client contacts AS with a request for service use through a secure channel (e.g. Secure Socket Layer, SSL). AS replies with a random challenge *nonce*. The client signs the concatenation (marked as $|$) of the *nonce* and a *commitment* of the form $commitment = g_2^{rand} \bmod n$, where the *rand* is a secret value chosen by the client. The signature is sent to AS which replies with its signature on the *commitment*. As a result the client committed to the value *rand* and received a signature on that commitment so he must use the value in future steps because PA will demand the AS signature. We use standard RSA signatures in our implementation. The *rand* value is crucial for security so it cannot be released by the client anywhere.

The client creates the token with the generator g_1 and the *rand* value as a secret exponent. The token is of the form $C = g_1^{rand} \bmod n$, $A = g_1^a \bmod n$ where a is a random number $\in_R \mathbb{Z}_p$. Client also creates NIPDLE between C and *commitment* to be able to prove the correct construction of the token to PA (he needs to convince PA that the exponent in the *commitment* and the token is the same one).

The client anonymously sends the token and the encrypted proof of its construction to PA using anonymous routing (e.g. [11]). PA verifies the construction and runs the PKDL protocol with an encrypted response z to find out whether the client knows the secret *rand*. The PKDL run is stored for future disputes as it can be used for an identity revelation. If everything is fine, PA signs the token and sends the signature to the client. This signature certifies that the token is of a correct form and that it was firstly spent at PA (Σ-protocol was run there). Now the token and PA's signature on it can be transferred to the smartcard. The authentication phase follows.

Authentication Phase

The authentication phase is a simple PKDL protocol with an encrypted response z. The access control device verifies PA's signature. If the signature is fine the device is convinced that the token is of a correct form and that it was used once before. If the PKDL protocol runs fine the client is accepted for the service use.

Disputes

In the case of rule breaking the token owner identity must be revealed. This could be done by the cooperation of PA and AS, because a Σ-protocol was run with both of these entities. It comes from the very basic property of the Σ-protocol that the secret exponent *rand* can be revealed from 2 different protocol runs. The *rand* computation is based on (1), where values without comma come from the PKDL with PA and values with comma come from the authentication phase PKDL with an access control device.

$$rand = \frac{z - z'}{e - e'} \bmod n. \tag{1}$$

Such a revelation is only available if PA and AS cooperate and because PA should be an authority driven e.g. by a state or court there is an assumption that service providers

are not able to break user anonymity unless they have a solid evidence for rule breaking. In addition no entity (not even PA) at all is able to abuse its rights to break user anonymity alone.

4. Scheme Analysis

The two key requirements for the scheme were security and efficiency. We provide analysis of these areas in this section.

4.1 Security Analysis

We discuss all the requirements stated in Section 2 here:

1. Soundness and Completeness: authentication is successful only if the PKDL protocol in the authentication phase is accepted. That is why the Soundness and Completeness come directly from the Σ-protocols used as the building blocks. The only entity knowing a valid *rand* is a valid user. Others know only a *commitment* to it which cannot be used for a *rand* extraction under the Discrete Logarithm problem assumption.

2. Anonymity: the identity is hidden in the token in the form of commitment which is unbreakable under the Discrete Logarithm assumption. The attacker would be successful if he was able to compute $w = \log_g c$ in the chosen group which is considered to be hard for big numbers (comparable to e.g. RSA implementation). Σ-protocols release no extra information based on the HVSZK property.

3. Spread Traceability: the identity can be revealed from the token but only if PA and AS cooperate. This feature works thanks to the Special Soundness property of Σ-protocols. The property guarantees that the secret exponent, which contains the user identity, is opened in the case of at least two protocol runs.

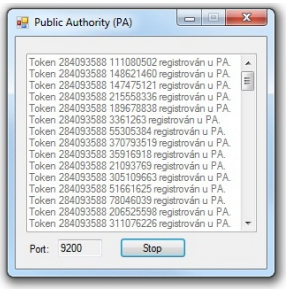
4. No Framing: the user commits to a secret, unique *rand* value in the registration phase. As he is the only one knowing that value, no one else can frame him by using it. The *rand* value should be therefore unique and never re-used.

4.2 Efficiency Analysis

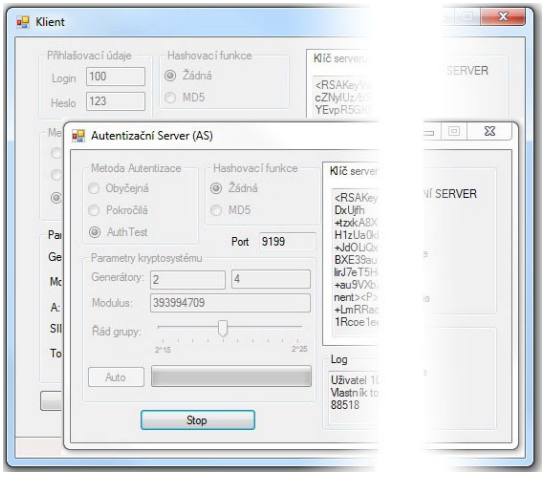
The whole scheme is divided into two parts. The first one (registration) is run in a computer network among computers. Efficient 3-way protocols are used and computations are done locally. The second phase (authentication) runs between a smartcard and an access device. The tokens are stored in a smartcard memory. With an average card memory (100kB) hundreds of tokens can be stored at once. This phase requires only one multiplication and addition on the card side which we consider to be efficient. This state is achieved thank to the communication-computation tradeoff. The authentication phase involves only 3 messages between sender's card and an access control device. That is why we find the scheme usable with RFID cards which do not provide as secure and reliable communication interface as standard contact cards do. In comparison to most of other schemes our solution is independent on a user group size.

4.3 Implementation

The scheme presented in this paper has been implemented in a testing .NET environment (see Figure 5). Although it works only as a proof of concept it certifies to the possibility of a real-world implementation which is our next goal. We expect to use .NET smartcards on the client side as also all other entities are built in .NET. The implementation allows anonymous authentication of the client entity by the authentication server entity. A newly introduced public authority entity has been also implemented to gather information needed for further malicious user revelations. The software allows to choose the order of the group used by cryptographic primitives. You can see all three key entities (Client, Authentication Server and Public Authority) implemented in Fig. 5. The server and client windows are trimmed due to readability.



(a) The Public Authority



(b) The Authentication Client and Server

Fig. 5. The testing implementation of the scheme.

5. Conclusion and Future Plans

We presented an anonymous authentication scheme for RFID smartcards in this paper. Our system provides anonymity for electronic device users as well as detection of malicious users for service providers. As a result the system can be used by both users requiring privacy and service providers requiring user responsibility. There is no need for a trusted entity or trusted HW in the scheme so it could be used in environments without trust. The cryptographic core is based on established primitives like Σ-protocols, one-time signatures and RSA signatures therefore security of the scheme could be easily verified.

According to future plans we would like to improve the zero-knowledge property. Although we provide security thanks to the Σ-protocol with an encrypted response there is probably a better way as the family of Σ-protocols can be efficiently converted [9] to Zero-Knowledge Proofs of Knowledge. We would like to use this feature to create a scheme with even stronger security assumptions.

Acknowledgements

Sponsored under the National Program of Research II by the Ministry of Education, Youth and Sports of the Czech Republic in 2C08002 Project - KAAPS Research of Universal and Complex Authentication and Authorization for Permanent and Mobile Computer Networks.

The author is a holder of the Brno Stipend for Talented Doctoral Students.

References

[1] BRANDS, S. Untraceable off-line cash in wallets with observers. In *Proceedings of the 13th Annual International Cryptology Conference on Advances in Cryptology*. Santa Barbara (USA), 1993, p. 302 – 318.

[2] BURMESTER, M., DE MEDEIROS, B., MOTTA, R. Robust, anonymous rfid authentication with constant key-lookup. In *Proceedings of the 2008 ACM Symposium on Information, Computer and Communications Security*. Tokyo (Japan), 2008, p. 283 – 291.

[3] BELENKIY, M., CAMENISCH, J., CHASE, M., KOHLWEISS, M., LYSUANSKAYA, A., SHACHAM, H. Randomizable proofs and delegatable anonymous credentials. In *Advances in Cryptology - CRYPTO 2009*. Santa Barbara (USA), 2009, p. 108 – 125.

[4] CAMENISCH, J., HOHENBERGER, S., LYSYANSKAYA, A. Compact e-cash. In *24th Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Aarhus (Denmark), 2005, p. 302 – 321.

[5] CAMENISCH, J., HOHENBERGER, S., LYSYANSKAYA, A. Balancing accountability and privacy using e-cash. In *Security and Cryptography for Networks, 5th International Conference SCN 2006 Proceedings*. Maiori (Italy), 2006, p. 141 – 155.

[6] CAMENISCH, J., STADLER, M. Efficient group signature schemes for large groups. In *Advances in Cryptology — CRYPTO'97. 17th Annual International Cryptology Conference Proccedings.* Santa Barbara (USA) 1997, p. 410 – 424.

[7] CHAUM, D., HEYST, E. V. Group signatures. In *Advances in Cryptology — EUROCRYPT'91. Workhsop on the Theory and Application of Cryptographic Techniques.* Brighton (UK), 1991, p. 257 – 265.

[8] CRAMER, R. *Modular Design of Secure, yet Practical Cryptographic Protocols*. PhD thesis. Amsterdam (The Netherlands): University of Amsterdam, 1996.

[9] CRAMER, R., DAMGARD, I., MACKENZIE, P. Efficient zero-knowledge proofs of knowledge without intractability assumptions. In *Public Key Cryptography. Third International Workshop on Practice and Theory in Public Key Cryptosystems PKC 2000*. Melbourne (Australia), 2000, p. 354 – 373.

[10] FURUKAWA, J., IMAI, H. An efficient group signature scheme from bilinear maps. In *Information Security and Privacy. 10th International Conference ACISP 2005*. Brisbane (Australia), 2005, p. 455–467.

[11] GOLDBERG, I. On the security of the tor authentication protocol. In *Proceedings of the Sixth Workshop on Privacy Enhancing Technologies*. Cambridge (UK), 2006, p. 316 – 331.

[12] IBRAHIM, M. H. Resisting traitors in linkable democratic group signatures. *International Journal of Network Security*, 2009, vol. 9, no. 1, p. 51 – 60.

[13] KIAYIAS, A., YUNG, M. Group signatures with efficient concurrent join. In *Proceedings of EUROCRYPT '05. Annual International Conference on the Theory and Applications of Cryptographic Techinques.* Aarhus (Denmark), 2005, p. 198 – 214.

[14] KIM, S., RHEE, H. S., CHUN, J. Y., LEE, D. H. Anonymous and traceable authentication scheme using smart cards. In *International Conference on Information Security and Assurance ISA 2008*. Busan (Korea), 2008, p. 162 – 168.

[15] LI, X.-X., QIAN, H.-F, LI, J.-H. Democratic group signatures with threshold traceability. *Journal of Shanghai Jiaotong University (Science)*, 2009, vol. 14, no. 1, p. 98–101.

[16] LINDEL, A. Y. *Anonymous Authentication.* [online] Cited 2010-01-04. Available at: http://www.aladdin.com/blog/pdf/Anonymous-Authentication.pdf.

[17] POPESCU, C. An efficient id-based group signature scheme. *Studia Univ. Babes-Bolyai, Informatica*, 2002, vol. 47, no. 2, p. 29 – 38.

[18] SANDER, T., TA-SHMA, A. Auditable, anonymous electronic cash extended abstract. In *Proceedings of the 19th Annual International Cryptology Conference on Advances in Cryptology*. Santa Barbara (USA), 1999, p. 555–572.

[19] SCHAFFER, M., SCHARTNER, P. Anonymous authentication with optional shared anonymity revocation and linkability. In *Seventh Smart Card Research and Advanced Application IFIP Conference CARDIS 2006.* Tarragona (Spain), 2006, p. 206 – 221.

[20] SCHNEIER, B. *Our Data, Ourselves*. [online] Cited 2009-12-16. Available at: http://www.wired.com/print/politics/security/commentary/securitymatters/2008/05/securitymatters_0515.

[21] SCHNEIER, B. *The Tech Lab.* [online] Cited 2009-12-16. Available at: http://news.bbc.co.uk/1/hi/technology/7897892.stm.

[22] SCHNORR, C. Efficient signature generation by smart cards. *Journal of Cryptology*, 1991, vol. 4, no. 3, p. 161 – 174.

[23] SHAHANDASHTI, S. F., SAFAVI-NAINI, R. Threshold attribute-based signatures and their application to anonymous credential systems. In *Progress in Cryptology – AFRICACRYPT 2009. Second International Conference on Cryptology in Africa.* Gammarth (Tunisia), 2009, p. 198 – 216.

About Author...

Jan HAJNY works as an academic assistant at the Faculty of Electrical Engineering and Communications at Brno University of Technology. He is focused on cryptography and its application in computer network authentication. He is also involved in the research of an universal authentication device.